

IN THE CLAIMS

Claims 1-26 (Cancelled)

27. (currently amended) A software ensemble stored on a computer readable medium and executable by a computer ~~in a software system that includes a plurality of software units stored on the computer readable medium~~, the software ensemble unit comprising:

a plurality of software units, each software unit of said plurality of software units including a method and data;

an executive software unit including links between said plurality of software units and between software units of the plurality of software units and an outside of the software ensemble.

~~an output gate for transmitting a message to invoke a method at one or more of the plurality of software units; and~~

~~a variable accessed at runtime that identifies the one or more of the plurality of software units that is to receive the message and identifies the method that is to be executed at the identified one or more of the plurality of software units.~~

28. (currently amended) A software ensemble~~unit~~ according to claim 27, a first software unit of the plurality of software units further comprising an input gate for receiving a message from another software unit of the plurality of software units to invoke a method at the first software unit, the first software unit and the method being determinable ~~by the other software unit~~ at runtime of the other unit via the links.

29. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein a first software unit of the plurality of the reusable software units is interconnected by the executive software unit to the one or more of the plurality of software units via one or more channels.

30. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein a first the software unit of the plurality of software units does not include an absolute address of the one or more of the plurality of software units ~~that are unknown to the software unit at its state.~~

C
|
31. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein a first said reusable software unit of the plurality of software units does not have knowledge of the name of unknown methods that areis to be executed at the one or more of the plurality of software units, ~~that are unknown to the software unit at its state.~~

32. (cancel)

33. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein the software unit of the software ensemble is described by a model M , given by:

$$M = (inGates, \{inSign_g\}, \{a_g\}, Q, q_0, outGates, \{outSign_{gt}\}, \{outFunction_{gt}\}),$$

where $inGates$ is the set of software unit input gates, $outGates$ is the set of software unit output gates, a_g is an action for every g in $inGates$, Q is the set of software unit states, q_0 is the software unit's initial state, $inSign_g$ is the gate gt input-output signature for each gt

in $outGates$, and $outfunction_{gt}$ is the gate gt output function for every gt in $outGates$, where the signature is a 2-tuple containing the range set of incoming and outgoing parameters, and the output function of gate gt maps a set of incoming values into a value belonging to the input signature of the output gate gt .

34. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein ~~two or more of the plurality of software units are combined into a the~~ software ~~unit~~ ensemble, E , is defined by:

$E = (inGates, \{inSign_g\}, \varepsilon, M_\varepsilon, outGates, \{outSign_{gt}\}, \{outFunction_{gt}\})$, where
 ε is the ensemble executive that keeps structure of the ensemble, and M_ε is the model of the ensemble executive, $inGates$ is the set of the ensemble software unit input gates, $inSign_g$ is the gate g input-output signature for each g in $inGates$, $outGates$ is the set of ensemble software unit output gates, $outSign_{gt}$ is the gate gt output-input signature for each gt in $outGates$, and $outFunction_{gt}$ is the gate gt output function for every gt in $outGates$, where a signature is a 2-tuple containing the range set of incoming and outgoing parameters, and the output function of gate gt maps a set of incoming values into a value belonging to the input signature of output gate gt .

35. (currently amended) The software ensemble~~unit~~ according to claim 34, where the model of the ensemble executive is defined as a model of a reusable software unit augmented with a structure function $\sigma: Q \rightarrow \Sigma^*$, where Σ is equal to $(C, \{M_c\}, L, \Xi)$, and C is the set of reusable software units that belong to the ensemble, M_c is the definition of each reusable software unit c , belonging to set C , L is a set of channels, and Ξ is the order function indicating a sequence in which actions are invoked.

36. (currently amended) The software ensemble~~unit~~ executive according to claim 34 35, wherein a channel is a 3-tuple defined by:

$$((i, g_i), (j, g_j), (dF, rF)),$$

where i is the name of a sender ~~source~~ software unit, g_i is a gate of i , j is a receiver software unit, g_j is a gate of j , dF is the channel direct filter and rF is the channel reverse filter, where a filter defines a transformation that is applied to the values communicated through a channel.

37. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein the plurality of software units can be located in: different threads, different processes, different processors and/or different computers.

38. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein several of the plurality of software units are concurrently active and access shared memory.

39. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein operations by the plurality of software units are performed synchronously and/or asynchronously.

40. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein the plurality of software units is stored in persistent storage.

41. (currently amended) The software ensemble~~unit~~ according to claim 40, wherein the persistent storage is one or more storage units selected from the group consisting of: a hard disk, a CDROM, a DVD, a floppy disk, and a magnetic tape.

42. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein the plurality of software units is enabled to run in a memory selected from the group consisting of: RAM, PROM, EPROM and EEPROM.

C
/ 43. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein the plurality of software units is serialized for storing and/or communication purposes.

44. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein structural inheritance is utilized to build new software ensembles ~~units~~ from existing software ensembles ~~units~~.

45. (currently amended) The software ensemble~~unit~~ according to claim 27, wherein objects may be utilized as software units having no output gates.

46. (currently amended) The software unit according to claim ~~27~~ 47, further including a message breaking mechanism that provides a realization of the variable mechanism by separating the message name from the message parameters.

47. (currently amended) A software ensemble~~unit~~ stored on a computer readable medium and executable by a computer ~~in a software system that~~

~~includes a plurality of software units stored on the computer readable medium, the software ensemble~~unit comprising:

a plurality of software units, each software unit including an output variable that provides access to other software units of the plurality of software units, the plurality of software units also including the software executive that manages a set of channels between the plurality of software units; and

one or more output gates, each output gate representing one message that can be sent to the output variable,

wherein the output variable, the one or more output gates and the set of channels determine one or more of methods invoked at the plurality of software units when a message is sent to the output variable of the software unit.

48. (new) The software ensemble according to Claim 47, wherein one or more of the methods is invoked at the one or more of the plurality of software units when a message is sent to the output variable of the invoking software unit.

49. (new) The software ensemble according to Claim 47, wherein the software ensemble is reconfigurable at runtime.

50. (new) The software unit according to Claim 47, wherein the software unit of the plurality of software units is interconnected to the one or more of the plurality of software units via one or more channels managed by the executive software unit.

51. (new): The software ensemble according to claim 34, wherein the input gate and the channels determine the one or more of the methods invoked at the one or more of the plurality of the software units when a message is received by the input gate of the software ensemble.

C
/

52. (new): The software ensemble of claim 35, wherein the output gate of a software unit and the channels determine the one or more of the messages to be sent through the output gates of the software ensemble.

53. (new): The software ensemble according to claim 27, wherein the software ensemble is configured as a module of another software ensemble.
